第6章 数据访问控制

成都信息工程大学 白杨 副教授

2024年X月X日

第6章 数据访问控制

本讲内容概要：

第6章 数据访问控制

本讲内容概要：

01 第一节—主客体身份标识与认证

02 第二节—权限管理

03 第三节—访问控制

04 第四节—数据访问控制实例

01 6.1主客体身份标识与认证

在数据访问控制中，主体和客体是两个关键概念。主体代表可以访问系统或资源的实体，比如用户、程序或设备。客体则是主体试图访问或操作的资源，比如文件、数据库记录或网络服务。

01 6.1.1 主体身份标识

1 身份验证和授权

系统通过比对主体提供的身份标识信息来验证其身份，确保合法主体才能访问系统资源。授权则依据主体的身份标识来确定其访问权限，限制主体的操作范围。

基于用户名/口令的验证

用户通过输入预设的用户名和口令来证明自己的身份

基于数字证书的验证

系统通过验证数字证书的合法性来确认主体身份。

基于生物特征的验证

系统通过识别主体的指纹、虹膜、声纹等生物特征来证明其身份。

基于令牌的验证

主体持有一次性密码令牌，通过验证令牌的有效性来确认身份。

01 6.1.1 主体身份标识

2 访问控制

主体身份标识是实施访问控制的关键要素。根据主体的身份标识信息来确定其访问权限，有效限制了主体对系统资源的访问范围。

01 基于角色的访问控制(RBAC)：根据主体所担任的角色来确定其访问权限

02 基于属性的访问控制(ABAC)：根据主体的动态属性(如部门、职位、项目等)来动态确定其访问权限

03 基于规则的访问控制(ABAC)：通过预设的访问规则来确定主体的访问权限，如时间、地点、设备等因素。

01 6.1.1 主体身份标识

3 审计和追溯

主体身份标识为审计和追溯提供了重要依据。系统可以记录主体的操作活动，并根据其身份标识信息关联到具体的主体。这样有助于在发生安全事件时及时识别责任主体，并进行事后分析和处置。同时，审计记录也可以用于检查系统访问策略的执行情况，持续优化安全管控。

01 6.1.1 主体身份标识

4 风险管理

通过确认主体身份，系统可以评估其风险特征，如权限、敏感操作等，并据此制定相应的安全策略。同时，主体身份标识还可以用于分析安全事件的成因，有助于评估和预防系统面临的风险。

01 对高风险主体(如超级管理员)实施更严格的访问控制和审计策略。

02 对某些敏感操作(如删除关键数据)设置双重身份验证等安全防护措施。

03 对异常身份标识行为(如非工作时间登录、登录地点异常等)及时预警并采取相应措施。

01 6.1.1 主体身份标识

5 信任传递

主体身份标识还支持跨系统的信任传递。当主体在不同系统中使用相同的身份标识时，这些系统可以相互验证主体的身份，实现单点登录、联合认证等功能，提高用户体验，同时也增强了整体的安全性。

01

同一个企业内部的OA系统、ERP系统、云存储系统等，都可以共享主体的身份标识信息，实现单点登录，使用户无需反复进行身份验证。

02

跨组织的信任传递也越来越重要。通过建立主体身份标识的对应关系，不同组织间的系统可以相互验证主体身份，实现跨域访问和协作。这对于供应链管理、跨境电商等场景非常有价值。

01

6.1.2 身份认证

身份认证是数据安全管理中的一个关键概念，它是通过验证主体提供的身份标识来确保其确实是它所声称的身份的过程。身份认证通常涉及提供凭据，如用户名和密码、生物特征识别、智能卡或其他身份验证因素。

1 信息秘密的身份认证

2 物理安全性的身份认证

3 生物学信息的身份认证

2 数字签名的身份认证

4 行为特征的身份认证

01

6.1.2 身份认证

基于信息秘密的身份认证

依赖于所拥有的东西或信息进行验证，可以分为口令认证、单向认证和双向认证。口令认证是最常见的身份认证方法之一，通常涉及用户提供一个用户名和密码来验证其身份。单向认证是指只有一个方向的认证，例如服务器验证客户端的身份，但客户端不验证服务器的身份。双向认证是指两个方向的认证，例如客户端和服务器都需要验证对方的身份。

基于物理安全性的身份认证

采用基于智能卡的身份认证机制，认证方要求一个硬件如智能卡，智能卡中往往存有秘密信息，通常是一个随机数。只有持卡人才能被认证。这种方法可以有效的防止口令猜测，但也存在一些严重的缺陷，例如系统只认卡不认人，而智能卡可能丢失，拾到或窃得智能卡的人很容易假冒原持卡人的身份。为了克服这个缺陷，可以采用认证方既要求用户输入一个口令，又要求智能卡的方法。

01 6.1.2 身份认证

基于生物学信息的身份认证

主要采用基于指纹识别的身份认证、基于语音识别的身份认证以及基于视网膜识别的身份认证等。

01 采样：生物识别系统捕捉到生物特征的样品，唯一的特征将会被提取并且转化成数字的符号存入此人的特征模板。

02 抽取特征：用户在需要验证身份时，与识别系统进行交互，设备提取用户的生物信息特征。

03 比较：用户的生物信息特征与特征模板中的数据进行比较。

04 匹配：如果匹配，则用户通过身份验证

01

6.1.2 身份认证

基于行为特征的身份认证

通过识别行为的特征进行验证。常见的验证模式有语音认证、签名识别等。利用签名实现的身份认证是属于模式识别认证的范畴，其过程也必然遵循模式识别的基本步骤。

基于数字签名的身份认证

使用数字签名技术来确认某个实体（例如个人、组织或设备）的身份。在数字签名中，发送者使用自己的私钥对信息进行签名，接收者则使用发送者的公钥来验证签名的有效性。确保消息的完整性、真实性和不可否认性，从而实现身份认证的目的。数字签名主要有3种算法：RSA签名、DSS签名和Hash签名。

第6章 数据访问控制

本讲内容概要：

01 第一节—主客体身份标识与认证

02 第二节—权限管理

03 第三节—访问控制

04 第四节—数据访问控制实例

02 6.2 权限管理

确定哪些主体可以访问哪些资源以及以何种方式访问这些资源。在数据安全领域，权限管理根据系统设置的安全规则或策略来控制用户对资源的访问和操作，以确保系统的安全性和数据的保密性。权限管理包括用户认证和授权两部分。

01 用户认证：
用户合法身份的校验，只有合法的用户才能访问系统。

02 授权：
用户必须具有该资源的访问权限才能访问该资源。确保只有经过授权的用户才能访问特定的资源，并限制他们对这些资源的操作权限。

02 6.2.1 认证与授权流程

权限管理的认证与授权流程步骤：

用户身份认证：用户通过提供合法的身份标识（如用户名和口令、生物特征等）进行身份认证。

01

访问请求验证：系统验证用户提交的访问请求，检查其身份认证信息。

02

授权验证：系统根据用户的身份和权限策略，检查用户是否具有访问特定资源的权限。

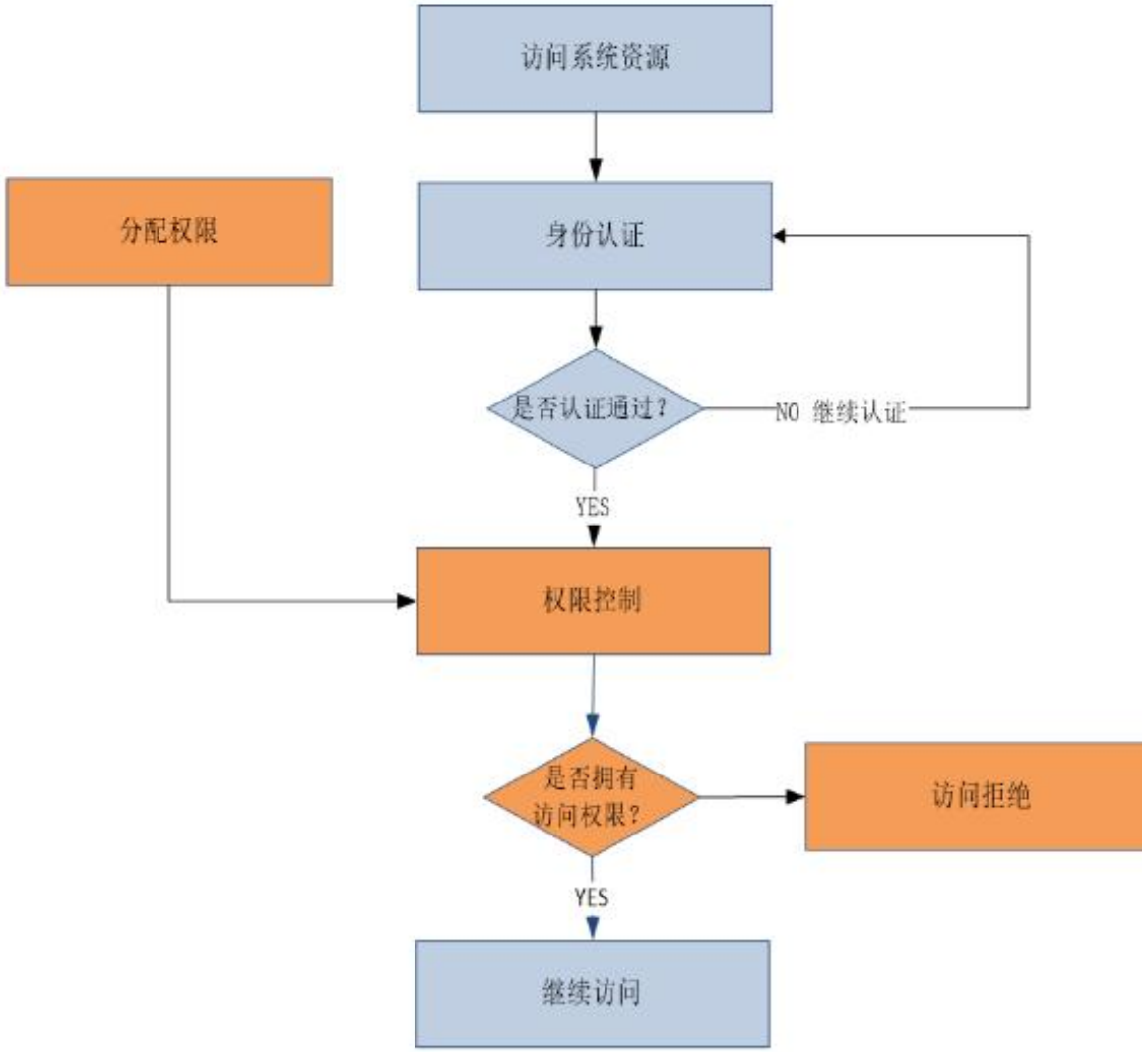
03

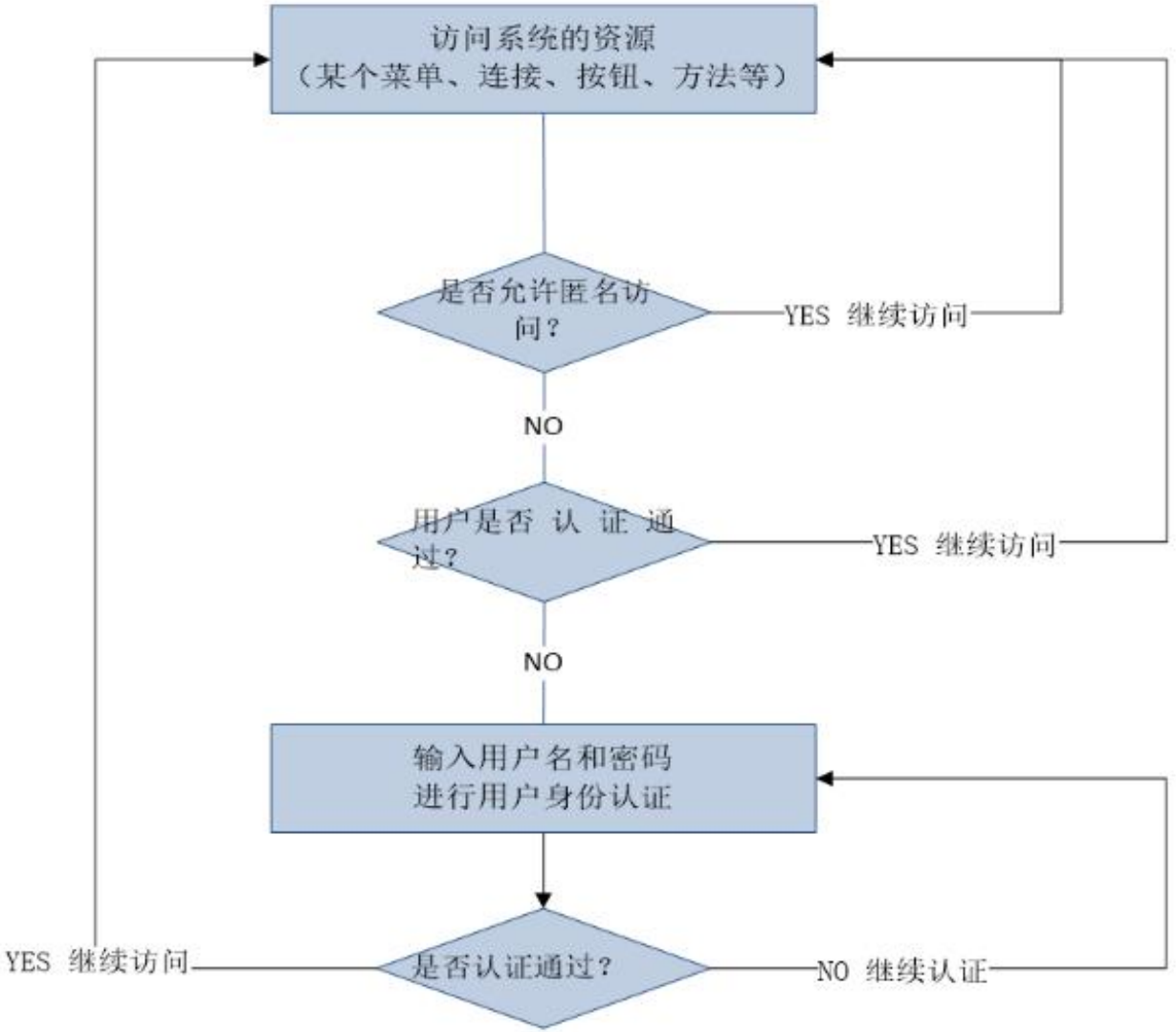
访问控制：如果用户被授权访问资源，则系统允许用户访问，并根据其权限限制其对资源的操作。

04

02 6.2.1 认证与授权流程

权限管理的认证与授权流程步骤：





02 6.2.2 权限管理的实践方法

- 权限管理的主要目的是确保只有经过授权的用户才能访问特定的资源，同时限制他们对这些资源的操作权限。权限管理通常包括访问控制、操作和数据控制、审计管理。
  - 访问控制：访问控制是权限管理的核心，它控制用户是否有权限访问某个资源。常见的访问控制方式包括：

01 强制访问控制（MAC）：基于系统定义的安全策略来控制对资源的访问

02 自主访问控制（DAC）：允许资源的所有者决定谁可以访问其资源以及以何种方式访问

03 角色基础访问控制（RBAC）：将权限授予角色而不是个体主体，并根据主体的角色确定其对资源的访问权限

02

6.2.2 权限管理的实践方法

操作和数据控制

系统可以控制用户是否有权限对某个资源进行特定的操作。操作控制通常是通过设置操作控制列表（OCL）来实现的。OCL是一种数据结构，它可以记录用户对资源的操作权限。例如，OCL可以记录用户是否有权限修改、删除或复制某个文件。

审计控制

记录用户的所有访问和操作记录，以便进行审计和追溯。审计记录包括用户的身份、访问时间、访问的资源以及执行的操作等信息，可以帮助系统管理员监控和审查用户的行为，及时发现潜在的安全问题

第6章 数据访问控制

本讲内容概要：

01 第一节—主客体身份标识与认证

02 第二节—权限管理

03 第三节—访问控制

04 第四节—数据访问控制实例

03 6.3 访问控制

访问控制是网络安全防范和保护的主要策略，其任务是保证网络资源不被非法使用和非法访问。

03 6.3.1 入网访问控制策略

入网访问控制是网络访问的第一层安全机制。它控制哪些用户能够登录到服务器并获准使用网络资源，控制准许用户入网的时间和位置。

1

用户名识别与验证：确定用户的身份。用户通过输入其用户名（或标识符，如电子邮件地址）来表明他们是谁，服务器将验证所输入的用户名是否合法。

2

用户口令识别与验证：系统通常会要求用户输入密码或其他形式的凭证（如生物识别信息、令牌等），确保只有拥有正确凭证的用户才能继续访问过程。

3

用户账户默认权限检查：检查用户的默认权限设置，这些设置定义了用户在网络中可以执行哪些操作。

03 6.3.2 操作权限控制策略

- 对用户在系统或网络中执行的各种操作进行权限分配和管理的策略。通过设定不同的操作权限，可以控制用户对系统资源的访问和操作，从而保障系统的安全性和稳定性。

- 01 角色划分：根据用户的职责和需求，将用户划分为不同的角色，并为每个角色分配相应的操作权限。
- 02 权限分配：为每个用户分配可执行的操作权限，如读取、写入、修改等，确保每个用户只能执行其职责范围内的操作。
- 03 权限审核：定期对用户的操作权限进行审核和调整，确保权限分配与用户的实际职责保持一致。
- 04 权限审计：记录用户的操作行为，以便在发生安全事件时进行追溯和调查。

03 6.3.3 目录安全控制策略

对系统或网络中的目录结构进行安全管理的策略。通过设定目录的访问权限和操作权限，可以控制用户对目录及其内部文件和子目录的访问和操作。

01 目录权限设置：为不同的目录设置不同的访问权限和操作权限，如读取、写入、创建、删除等。

02 目录继承：允许目录的权限设置继承自其父目录，简化权限管理。

03 特殊权限控制：对包含敏感信息的目录进行特殊权限控制，如限制访问用户、设置更严格的权限审核等。

04 目录访问审计：记录用户对目录的访问行为，以便进行安全审计和追溯。

03 6.3.4 属性安全控制策略

属性安全控制策略是指对系统或网络中的资源（如文件、目录、设备等）的属性进行安全管理的策略。通过设定资源的属性（如访问权限、加密属性、审计属性等），可以增强资源的安全性。

1. 属性定义：
为资源定义一系列安全属性，如访问权限、加密要求、审计日志等。

2. 属性分配：
根据资源的敏感程度和访问需求，为资源分配相应的安全属性。

3. 属性变更控制：
对资源属性的变更进行严格控制，确保只有授权用户才能修改资源的属性。

4. 属性审计：
记录资源属性的变更情况，以便进行安全审计和追溯

03 6.3.5 网络服务器安全控制策略

- 网络服务器安全控制策略是指对网络服务器进行安全管理的策略。通过设定服务器的安全配置、访问控制、日志审计等措施，可以保障服务器的安全性和稳定性。

01 安全配置：
对服务器进行安全配置，如设置强密码、关闭不必要的服务、禁用默认账户等。

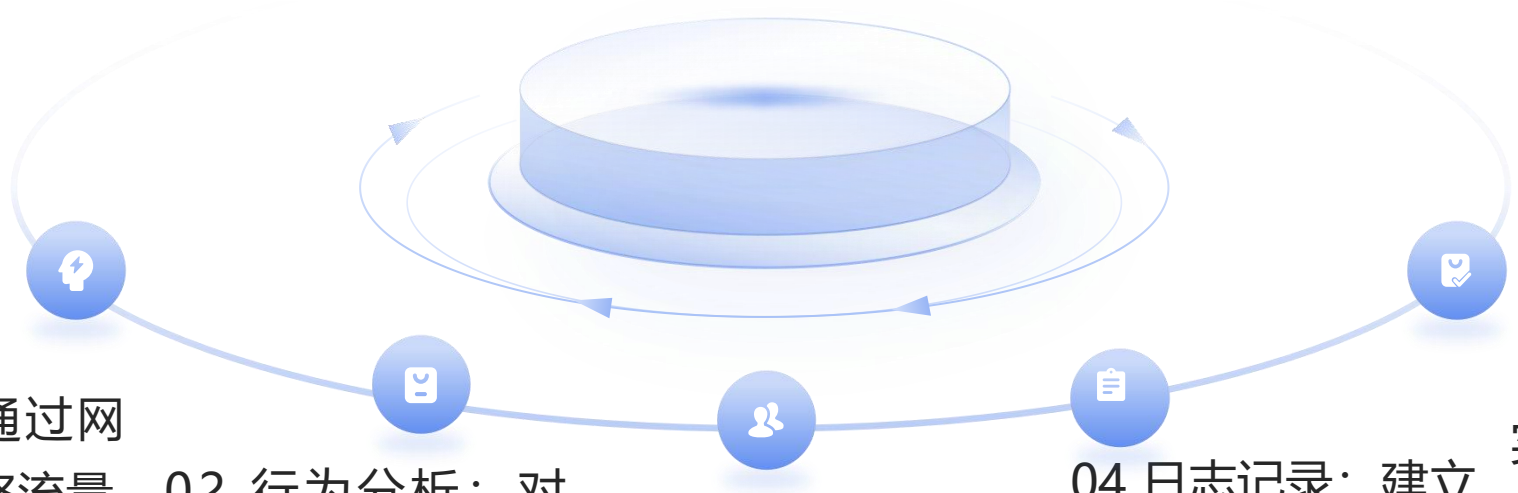
02 访问控制：
设定服务器的访问控制策略，如IP地址过滤、端口限制、访问权限分配等。

03 日志审计：
记录服务器的运行日志和访问日志，以便进行安全审计和追溯。

04 安全更新：
定期更新服务器的操作系统、应用程序和安全补丁，以修复已知的安全漏洞。

05 备份与恢复：
制定服务器的备份与恢复策略，确保在发生安全事件时能够迅速恢复服务器的正常运行。

03 6.3.6 网络监测和锁定控制策略



01 实时监控：通过网络监控系统对网络流量、登录情况、外部连接等进行实时监控，及时发现异常行为和安全事件。

02 行为分析：对监控到的数据进行分析，识别出异常或可疑的网络活动，如未经授权的访问、数据泄露等。

03 锁定控制：一旦发现异常行为，立即采取锁定措施，如阻断可疑连接、隔离受感染的设备等，以防止威胁进一步扩散。

04 日志记录：建立完善的安全日志管理机制，记录网络安全事件的发生、处理过程和结果，以便后续分析和审计。

05 实时警报：建立实时的网络安全警报系统，对网络异常行为进行监测和报警，提高应急处理的速度和准确性。

03 6.3.7 防火墙控制策略

防火墙控制策略是网络安全的第一道防线，它通过制定和执行一系列安全规则，对进出网络的数据包进行过滤和控制，从而保护内部网络免受外部威胁的侵害。

01 包过滤技术

根据设定的规则，检查数据包的源IP地址、目的IP地址、端口号等，对经过防火墙的数据包进行检查和过滤。

02 状态检测技术

以流量为单位对数据进行检测和转发。

03 访问控制

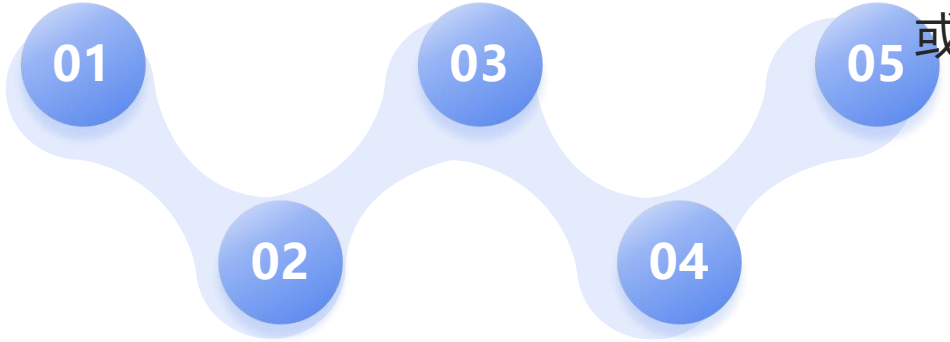
制定访问控制策略，对进出网络的数据流进行细粒度的控制。

04 日志记录和审计

记录所有经过的数据包信息以及安全事件的处理情况，以便后续进行日志分析和审计。

05 策略优化和更新

防火墙的控制策略也需要不断优化和更新，包括删除陈旧或冗余的规则、关注可能影响设备性能或安全性的重叠规则等。



第6章 数据访问控制

本讲内容概要：

01 第一节—主客体身份标识与认证

02 第二节—权限管理

03 第三节—访问控制

04 第四节—数据访问控制实例

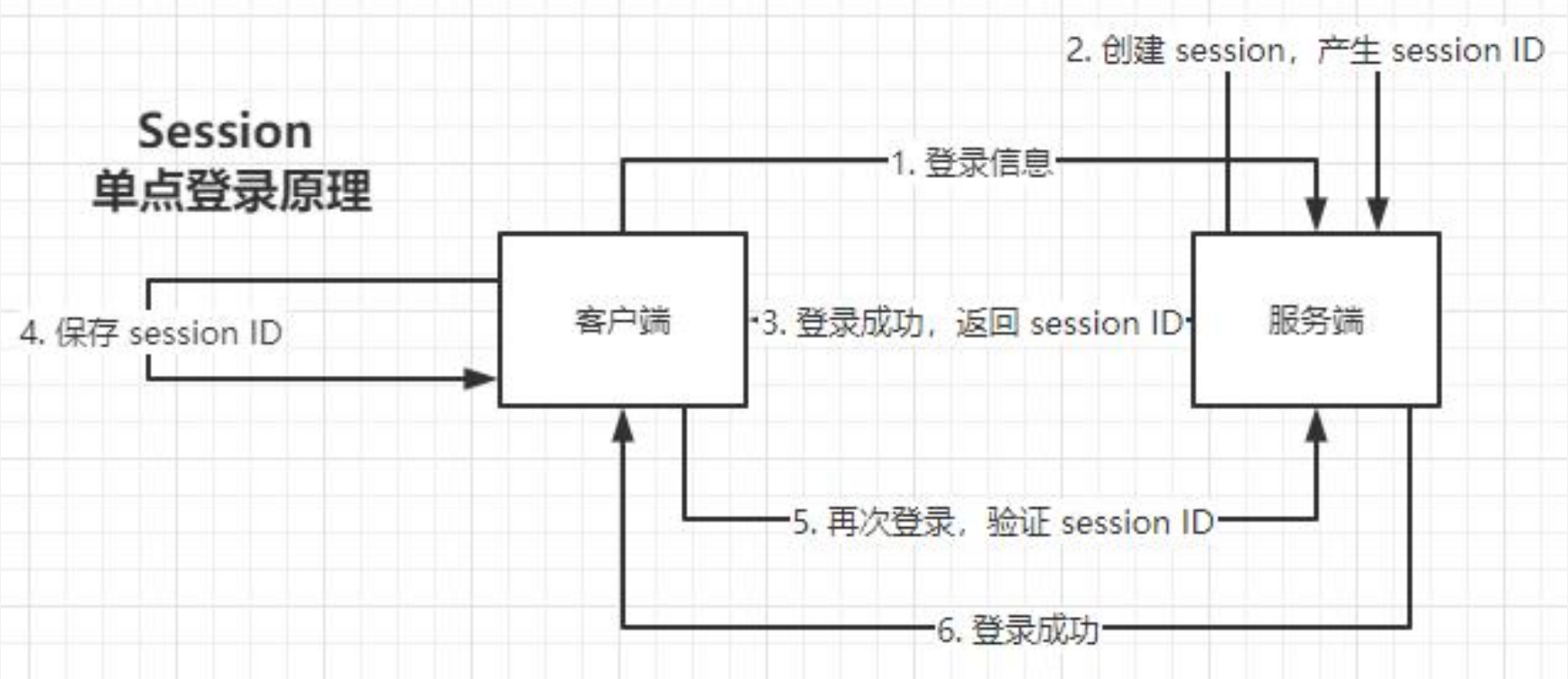
04 6.4.1 单点登录

单点登录(Single Sign-On，SSO)是一种用户只需一次登录，就可以访问多个应用系统的技术。单点登录的实现方案一般包含：Session 验证、Cookie验证、Token验证

04 6.4.1 单点登录

- 通过 Session 进行验证

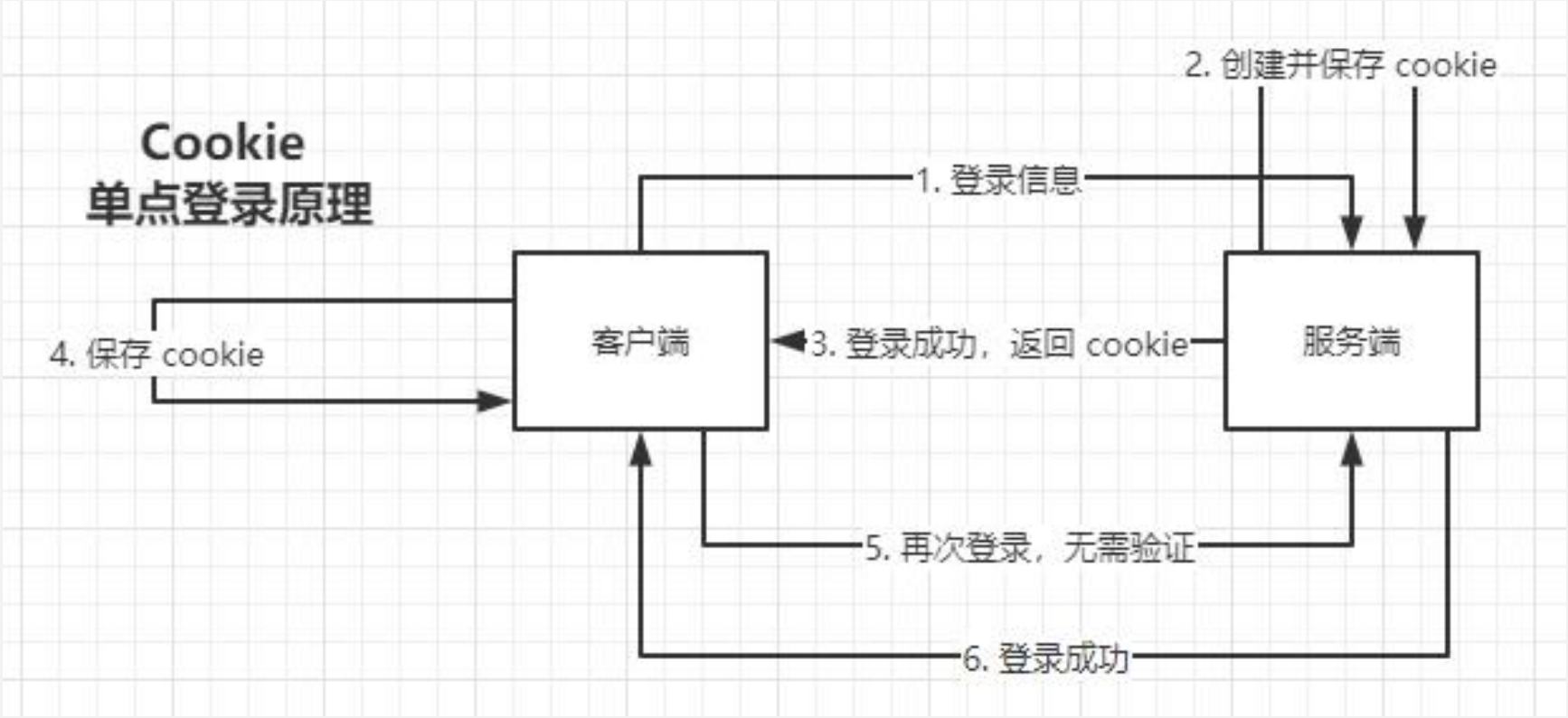
Session对象存储特定用户会话所需的属性及配置信息。用户在登录了一个系统后，服务器会将登录信息储存在一个Session中，产生Session ID，客户端会保存该 ID；当这个用户再登录其他系统时，服务器会自动复制上一个模块的Session该服务器的Session中，以获取用户登录信息，实现用户只登录一次，就可以登录其他系统。在用户退出登录时，服务器会自动删除Session。



04 6.4.1 单点登录

- 通过Cookie进行验证

  Cookie是某些网站为了辨别用户身份，由服务端生成，发给客户端暂时或永久保存的信息。例如，当我们打开一个网站，比如新浪、CSDN、知乎时，输入用户名和密码登录后系统会弹出是否保存Cookie，如果我们选择保存，在下一次登录时，就不需要再次输入用户名和密码，而是默认登录成功，直接进入页面。



通过Token进行验证

Token就是一种凭证，用户在登录注册时需要获取凭证，在经过验证后，方可登录相关被授权的应用。流程如下：

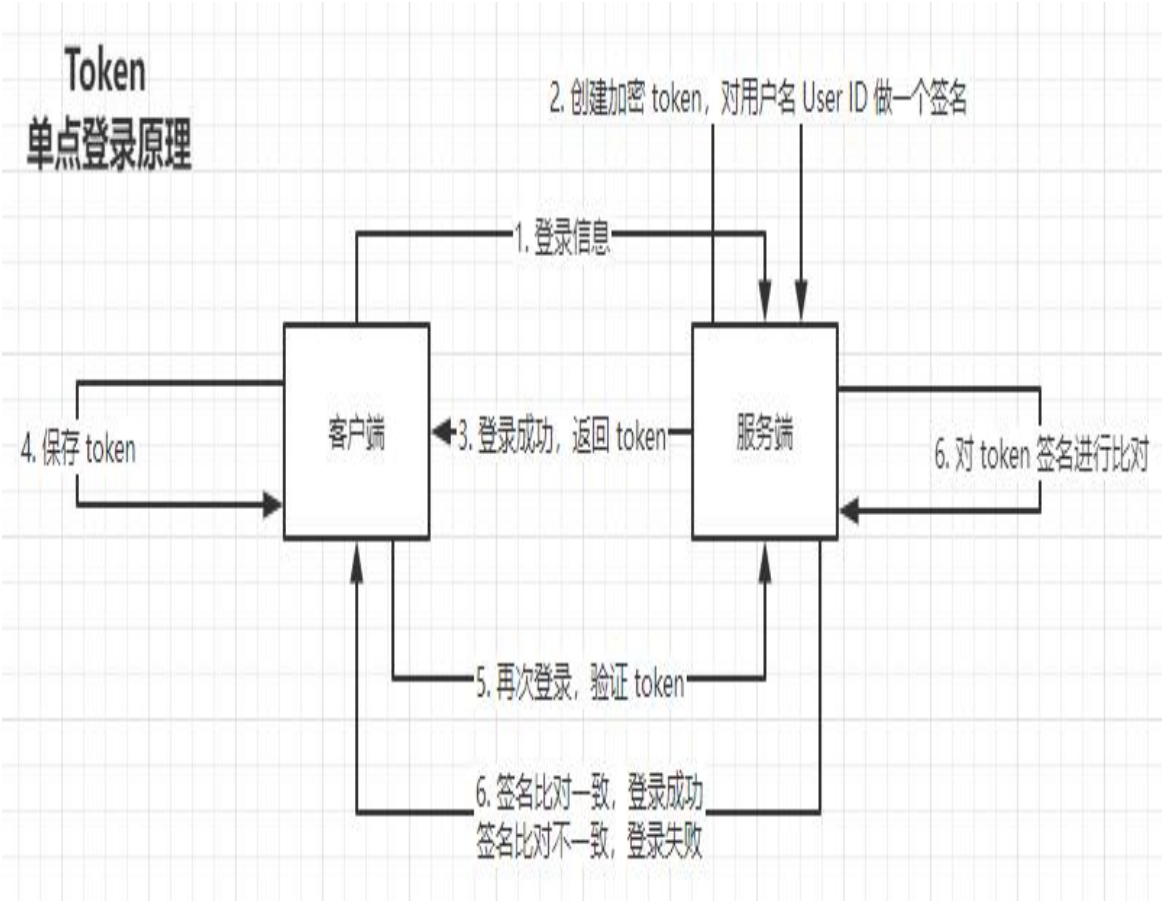
1. 用户在首次登录系统时输入账号和口令，服务器会收到登录请求，然后验证是否正确

2. 服务器会根据用户信息，如用户 ID、用户名、密钥、过期时间等信息生成一个Token签名，然后发给用户

3. 用户验证成功后，返回Token；

4. 前端服务器收到Token后，存储到Cookie或Local Storage里；当用户再次登录时，会被服务器验证Token；

5. 服务器收到用户登录请求后，对Token签名进行比对：如果Token验证正确，用户登录成功；如果Token验证不正确，用户登录失败，跳转到登录页。



04 6.4.2 数据库访问控制

数据库访问控制（Database Access Control，DAC）用于确保只有授权用户才能访问数据库中的数据并执行操作。它通过一系列机制来限制用户对数据库资源的访问，例如：

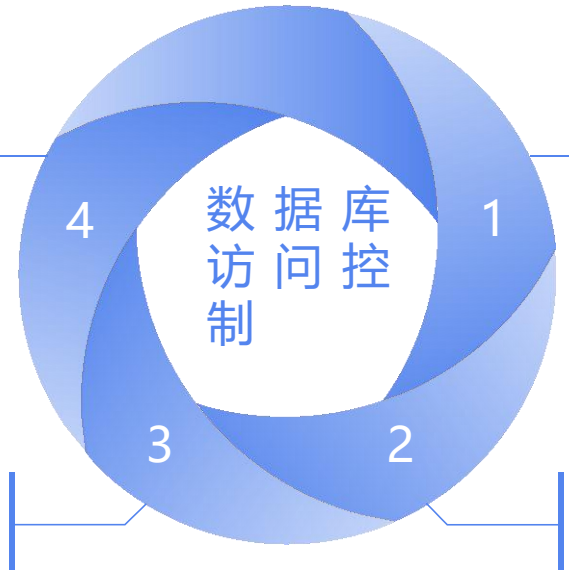
1. 身份验证：验证用户的身份，确保其拥有访问数据库的权限。
2. 授权：授予用户访问特定数据库资源的权限，例如表、视图、存储过程等。
3. 审计：记录用户的数据库操作，以便追踪和分析可疑行为。

04 6.4.2 数据库访问控制

数据库访问控制对于保护敏感数据和防止数据库滥用至关重要。它可以帮助企业降低数据泄露、数据损坏和恶意攻击的风险。

强制访问控制（MAC）：根据数据的敏感度和用户的安全级别授予访问权限。

基于属性的访问控制（ABAC）：根据用户的属性和请求的上下文授予访问权限。用户的属性可以包括其部门、职位、工作地点等。请求的上下文可以包括请求的时间、地点和目的等。

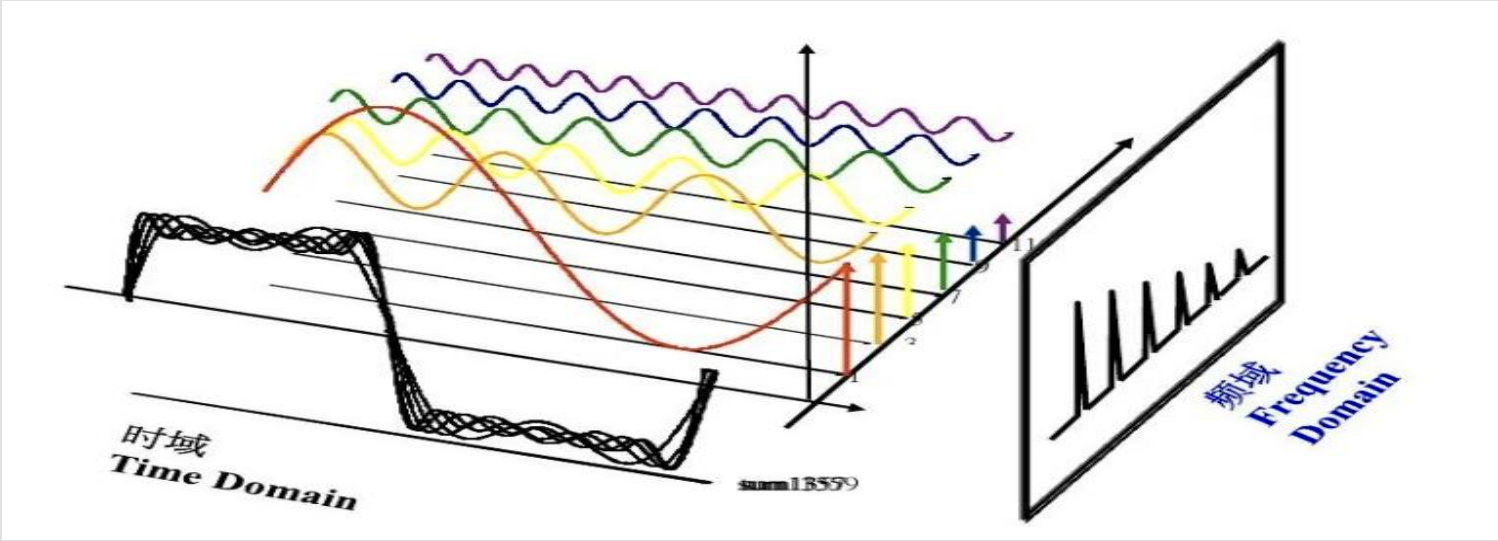


基于用户的访问控制（UBAC）：根据用户的身份授予访问权限。每个用户都有一个唯一的用户名和密码，并且只能访问其经过授权的数据库资源。

基于角色的访问控制（RBAC）：根据用户的角色授予访问权限。每个角色都有一组与之关联的权限，并且用户可以分配多个角色。

04 6.4.3 数据接口访问控制

目前市场上比较常见的两种 API 访问控制方案，分别是 OAuth 2.0 和JWT（JSON Web Token）。它们都用于在应用程序和服务之间进行安全身份验证和授权



Auth 2.0 在大数据平台中的应用步骤：

(1) 大数据平台作为 OAuth 2.0 的资源服务器，负责提供数据接口。

(2) 第三方应用作为 OAuth 2.0 的客户端，需要向大数据平台申请访问令牌(access token)。

(3) 用户在第三方应用上进行授权，大数据平台的认证服务器负责对用户进行身份验证，并根据用户的权限颁发相应的访问令牌。

(4) 第三方应用携带访问令牌来调用大数据平台的数据接口，接口服务器根据令牌的信息进行授权验证。

```
@RestControllerpublic class OAuthController
{ @GetMapping("/login/wenxin")public
RedirectViewwxLogin(HttpServletRequest request)
{String clientId = "your_client_id";String redirectUri =
"http://localhost:8080/login/oauth2/code/wenxin";String
wxLoginUrl =
"https://accounts.wenxin.com/o/oauth2/v2/auth"
+"?client_id=" + clientId +"&redirect_uri=" + redirectUri
+"&response_type=code"
+"&scope=email%20profile";return new
RedirectView(wxLoginUrl);}
@GetMapping("/login/oauth2/code/wenxin")public
String wxCallback(@RequestParam("code") String code)
{// 在此处通过 code 向微信请求访问令牌，并获取用户信息
return "微信 OAuth 2.0 登录成功";}}
```

基于 JWT 的数据接口访问控制步骤：

```
ontroller
lass AuthController {

Autowired
vate JwtTokenUtil jwtTokenUtil;
Autowired
vate UserService userService;

PostMapping("/login")
blic ResponseEntity<?> login(@RequestBody UserLoginRequest userReq) {
  // 在此处校验用户名和密码，验证成功后生成 JWT
  String token = jwtTokenUtil.generateToken(userService.loadUser(userReq.getUs
  return ResponseEntity.ok(new AuthResponse(token));


GetMapping("/user")
blic ResponseEntity<?> getUserInfo(@RequestHeader("Authorization") String jwtT
  String username = jwtTokenUtil.extractUsername(jwtToken);
  // 在此处根据用户名获取用户信息
  User user = userService.loadUserByUsername(username);
  return ResponseEntity.ok(user);
```

一．用户通过身份验证后，大数据平台的认证服务会颁发一个 JWT token 给客户端。

二．客户端在后续访问数据接口时，将 JWT token 携带在请求头中。

三．数据接口服务器验证 JWT token 的合法性，根据 token 中包含的用户信息和权限信息来决定是否允许访问。

谢谢！